# *Supplementary Material*

## A. Encoder and Transformer Details

**Visual Encoder.** Video Swin Transformer [7] is adopted as the visual encoder because of its effectiveness in extracting robust spatio-temporal features. Multi-stage visual features with spatial strides of {4,8,16,32} are used for segmentation, *i.e.*, the last three stages for cross-modal fusion and the first two stages for multi-granularity optimization. We resize the multi-scale vision-language features to the same resolution and use element-wise addition to integrate them into a single layer for conditional segmentation.

**Textual Encoder.** The pre-trained RoBERTa [6] is used to encode language expressions due to its proven performance in natural language processing tasks. Each expression is encoded into word features and sentence features.

**Transformer.** Deformable Transformer [13] with 4 encoder and decoder layers is used to encode vision-language features and predict instance embeddings due to its effectiveness and efficiency in capturing global pixel-level relations.

## B. Instance Matching Details

Our instance matching process follows the standard paradigm used by previous transformer-based methods for video segmentation [11, 1, 2, 3, 10, 12]. Specifically, we use $N$=5 learnable instance queries for prediction and apply the Hungarian algorithm [4] to select the best result. To achieve this, SgMg predicts patch masks $\mathcal{M}_P$, bounding boxes $\mathcal{B}$, and confidence scores $\mathcal{S}$ for each expression. Using the set of predictions $y = \{\forall y^i, i \in [1, ..., N]\}$, where $y^i = \{\mathcal{M}_P^{i,j}, \mathcal{B}^{i,j}, \mathcal{S}^{i,j}\}_{j=1}^{T}$, we compute the matching loss $\mathcal{L}_{match}$ for each query based on the ground truth $\hat{y}$ and employ Hungarian algorithm to find the best match that has the minimum loss. $\mathcal{L}_{match}$ lies in three parts:

$$\mathcal{L}_{match} = \lambda_{\mathcal{M}_P}\mathcal{L}_{\mathcal{M}_P} + \lambda_{\mathcal{B}}\mathcal{L}_{\mathcal{B}} + \lambda_{\mathcal{S}}\mathcal{L}_{\mathcal{S}} \tag{1}$$

where $\lambda$ denotes the coefficient to balance $\mathcal{L}_{match}$.

## C. Further Implementation Details

Our training settings follow [11, 5, 1]. The data augmentation includes random resize, random crop, random horizontal flip, and photometric distortion. The models are trained using AdamW [8] optimizer for 12 epochs during pre-training, and 6 or 9 epochs during main training depending on whether pre-training is used. During pre-training on RefCOCOs, we set the initial learning rates of 2.5e-6, 1.25e-5, and 2.5e-5 for the text encoder, visual encoder, and the rest of the model, respectively. The pre-training employs a single frame, with the learning rates decayed by a factor of 10 at the 8th and 10th epochs. In the main training, we freeze the text encoder, and the initial learning rates of 2.5e-5 and 5e-5 are adopted for the visual encoder and the rest, respectively. The learning rates are divided by 10 at the 6th and 8th epoch.

During inference, we perform clip-wise segmentation as in [11]. Specifically, we set the clip length equal to the number of video frames for Ref-YoutubeVOS and 36 for Ref-DAVIS17 to enable better spatio-temporal feature representation and efficiency. Notably, our approach can also perform frame-wise segmentation to achieve good performance according to the referring image segmentation results presented in the main paper.

## D. Conditional Patch Segmentation

We present the pseudo-code of our conditional patch segmentation process in Fig. A. To be specific, instance embeddings are employed to predict conditional patch kernels. The conditional patch kernels are reshaped to dynamic weights and bias, which form two point-wise convolutions. Finally, point-wise convolutions are used to segment vision-language features to obtain patch masks.

```
def cond_patch_segm(inst_embeds, visi_lang_feats):
    # inst_embeds: (B, C)
    # visi_lang_feats: (B, C, H/i, W/i)

    # predict conditional patch kernels:
    cond_patch_kernel = Linear(inst_embeds)
    # reshape to form two point-wise convolutions:
    weights, bias = Parameterization(cond_patch_kernel)

    # conditional patch segmentation
    f = visi_lang_feats
    for i, (w, b) in enumerate(zip(weights, bias)):
        f = Point_Conv(f, weight=w, bias=b, stride=1)
        if i < len(weights) - 1:
            f = relu(f)

    # patch_segm: (B, p^2, H/i, W/i)
    patch_segm = f
    return patch_segm
```

Figure A. Pseudo-code of conditional patch segmentation.

## E. Ablation of Spectral Convolutions in SCF

We replace the spectral convolutions in Spectrum-guided Cross-modal Fusion (SCF) with spatial convolutions or linear layers, which contain more parameters than ours. As shown in Table A, our SCF that operates in the spectral domain achieves the best performance.

| Module | $\mathcal{J}\&\mathcal{F}$ | Para. Num. (M) |
|---|---|---|
| SCF w/ Spatial Conv | 57.6 | 4.7 |
| SCF w/ Linear | 57.9 | 2.4 |
| SCF (Ours) | **58.9** | **2.4** |

Table A. Ablation of SCF with different operations.

## F. Additional Ablation Study Results

We remove CPK/MSO to fully evaluate the options in our approach. As shown in Table B, deleting CPK/MSO leads to a 0.5/1.5% accuracy drop.

| Module | $\mathcal{J}\&\mathcal{F}$ | FPS |
|---|---|---|
| SgMg w/o CPK | 58.4 | 65 |
| SgMg w/o MSO | 57.4 | 66 |

Table B. Ablation of CPK and MSO.

## G. Additional t-SNE Visualizations

To further demonstrate the presence of feature drift, we present additional t-SNE [9] visualizations in Fig. B. Specifically, we add the feature decoding process into the model, where the token embeddings of encoded features $\mathcal{F}_{vl}$ are decoded using the decoder in [11] to obtain $\mathcal{F}_{vl}^{d}$ for all frames in each video. By visualizing these embeddings with t-SNE, we observe that the token embeddings of $\mathcal{F}_{vl}$ and $\mathcal{F}_{vl}^{d}$ are separated into two distinct clusters. This indicates that the decoding process results in feature drift. However, the segmentation kernels struggle to perceive this drift during forward propagation since the kernels are predicted before the feature decoding.

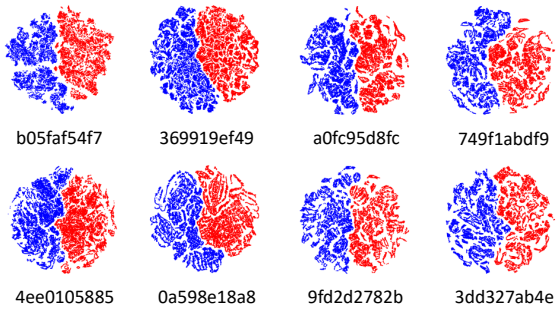


Figure B. t-SNE [9] visualization of the feature embeddings in different videos before (red cluster) and after (blue cluster) decoding.

## H. Additional Qualitative Results

In Fig. C, we present additional qualitative results that include occlusion, similar appearance, fast motion, and small objects.

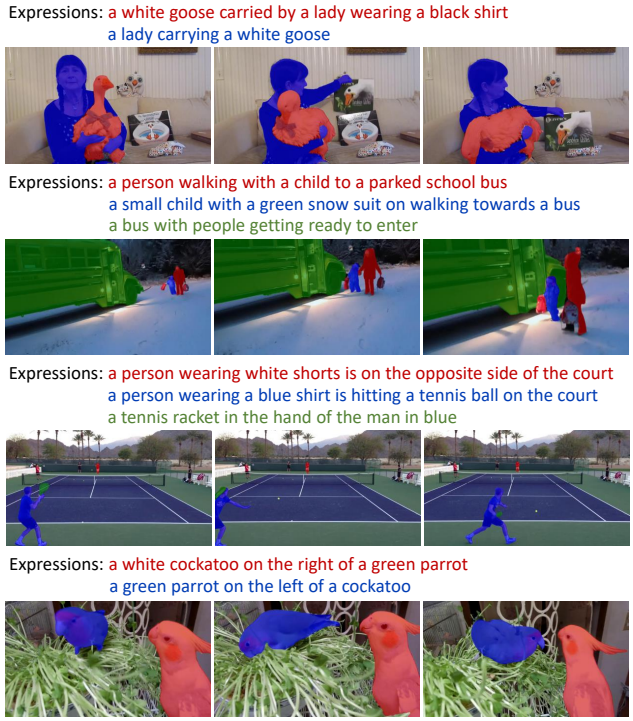


Figure C. Additional qualitative results of SgMg.

## References


[1] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, pages 4985–4995, 2022. 1

[2] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 1

[3] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Vita: Video instance segmentation via object token association. *arXiv preprint arXiv:2206.04403*, 2022. 1

[4] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1

[5] Dezhuang Li, Ruoqi Li, Lijun Wang, Yifan Wang, Jinqing Qi, Lu Zhang, Ting Liu, Qingquan Xu, and Huchuan Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In *AAAI Conference on Artificial Intelligence*, 2022. 1

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1

[7] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 1

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2

[10] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc.*

*IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8741–8750, 2021. 1

[11] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, pages 4974–4984, 2022. 1, 2

[12] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 588–605. Springer, 2022. 1

[13] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1